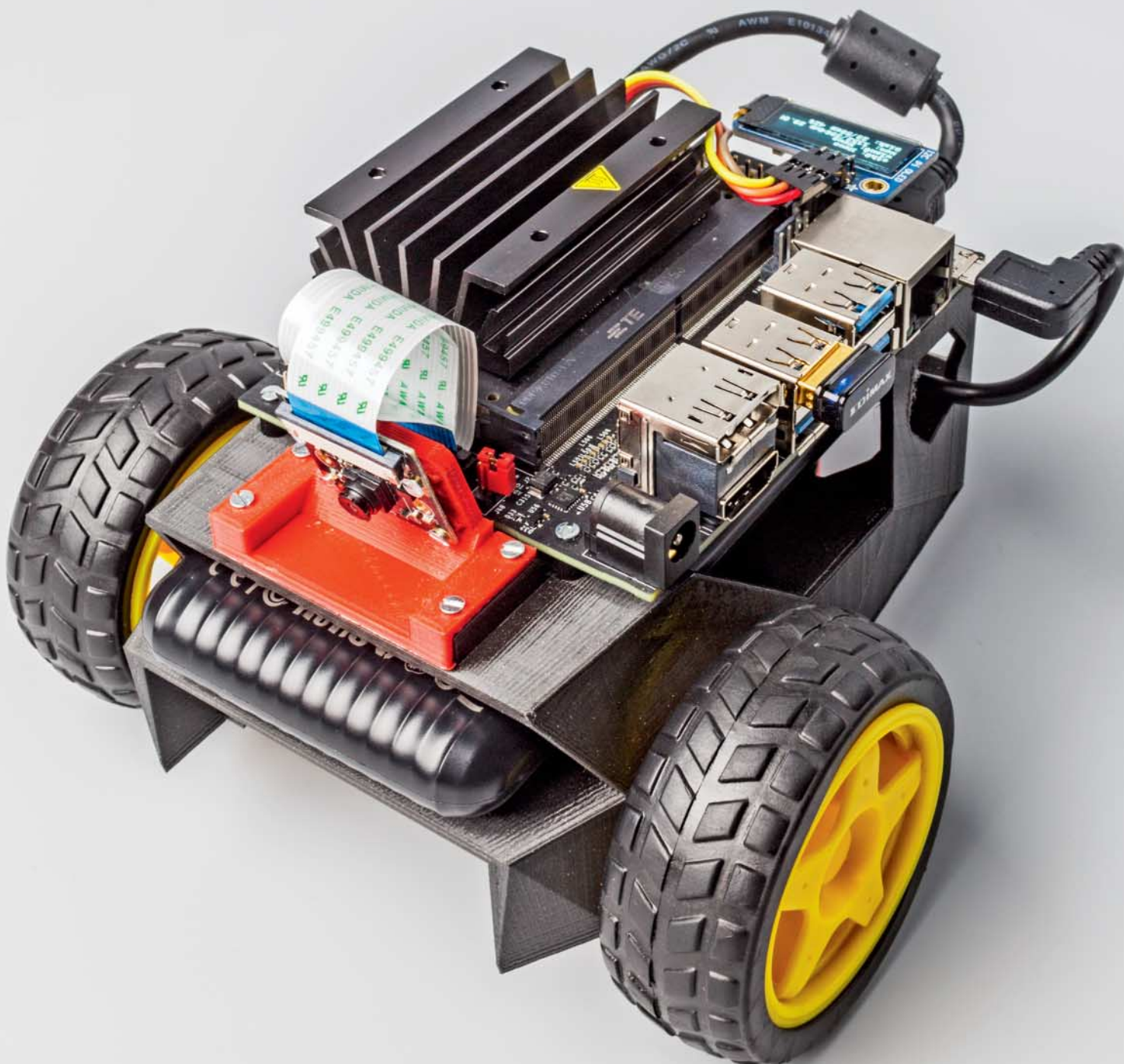


# JetBot

Der JetBot ist dank seines leistungsfähigen Prozessors nicht nur ein KI-fähiger Roboter, sondern mit Hilfe seiner Software auch ein Lernsystem. Und zwar ein recht komfortables und leicht verständliches.

von Heinz Behling



**R**oboter-Programmierung gilt als schwierig, in Zusammenhang mit Künstlicher Intelligenz (KI) sogar als für Laien kaum beherrschbar. Der Selbstbau-Roboter JetBot macht Schluss mit diesen Vorurteilen. Er ist nämlich nicht nur einfach zusammenzubauen, sondern bringt ein Software-Paket mit, das auch Einsteigern richtig Freude macht und schnell erste Erfolge erzielen lässt.

Seinen Namen verdankt er übrigens dem Steuercomputer, einem Jetson nano <sup>1</sup>. Dieser Computer enthält einen Grafik-Prozessor des Herstellers Nvidia, dessen zahlreichen Rechenkerne ideal sind für ein künstliches neuronales Netz. Das wird später benutzt, um Objekte zu erkennen. Der Jetson hat die Bauform eines Notebook-Speicherstreifens und sitzt auf einer kleinen Platine, dem Jetson nano Developer Kit. Das stellt die notwendigen Anschlüsse für USB, Kamera und Display sowie die Spannungsversorgung zur Verfügung.

Die Steuerung des JetBot erfolgt über eine Web-Oberfläche, die vom lokal auf dem Jetson laufenden Webserver bereitgestellt wird. Die Programmierung erfolgt in der Programmiersprache Python. Mitgeliefert werden eine Reihe von Beispielen, die in sogenannten Notebook-Dateien (die haben nichts mit tragbaren Computern zu tun, sondern eher mit Notizblöcken) Schritt für Schritt jeden einzelnen Befehl erklären. Der Clou: Die in den Notebooks enthaltenen Programmteile können auch gleich aus dem Anleitungstext heraus an den JetBot geschickt und dort ausgeführt werden. Sogar Änderungen sind in den Befehlen möglich und deren Auswirkungen können sofort überprüft werden. Doch dazu später mehr.

## Aufbau

Zunächst müssen Sie die Hardware des kleinen Roboters zusammenbauen. Auf github (Adresse dazu und Bezugsquellen für alle Teile siehe Kurzinfo-Link) können Sie die Projektdateien downloaden. Sie enthalten auch die 3D-Druckdateien für das Chassis des JetBot. Falls Sie keinen Zugang zu einem 3D-Drucker haben, lesen Sie den Textkasten „Universal-Chassis als Grundlage“ auf Seite 45.

Wenn Sie das Chassis selbst drucken, müssen Sie zunächst den Durchmesser der Antriebsräder wissen. Das in den Bezugsquellen angegebene Rad-Motor-Set hat 65mm-Räder. Dementsprechend müssen Sie die passenden Teile für das vordere Stützlager und den Kamerahalter drucken. Die Dateinamen weisen jeweils auf den Raddurchmesser hin. Falls Sie die falschen Dateien nehmen, steht der Roboter später schief und die Kamera guckt zu tief nach unten.

Über den Kurzinfo-Link finden Sie auch die Adresse des JetBot-Wiki. Es enthält eine

## Kurzinfo

- » KI-fähigen mobilen Roboter selbst bauen
- » Neuronales Netz zur Objekterkennung
- » Weitere Netz-Modelle

### Checkliste



#### Zeitaufwand:

zwei Stunden (ohne Zeit für 3D-Druck)



#### Kosten:

100 Euro zzgl. ca. 130 Euro für Jetson Nano Developer Kit



#### Löten:

einfache Lötarbeiten



#### Programmieren:

Python



#### 3D-Druck:

downloadbare Dateien mit PLA drucken

Alles zum Artikel  
im Web unter  
[make-magazin.de/xbat](https://make-magazin.de/xbat)

### Material

- » Jetson Nano Developer Set
- » Mikro-SD-Speicherkarte mit mindestens 32GB
- » 2 TT-Getriebemotoren mit Rädern 65mm breit
- » Kameramodul LI-IMX219-MIPI-FF-NANO
- » Motortreiber ADA2927
- » Powerbank 10000mAh Ausgang 5V/3A (Maße beachten: 138 × 68 × 15mm, siehe Bezugsquellen)
- » PiOLED-Display I<sup>2</sup>C, 128 × 32 Pixel
- » 2 USB-Kabel USB-A- auf Mikro-USB-Stecker, gewinkelt, 22cm lang
- » PIN-Leiste gewinkelt 90°
- » USB-Dongle Edimax EW-7811Un
- » Schrauben und Muttern M2, M3
- » Rollkugel 1 Zoll
- » Druckfilament für Chassis

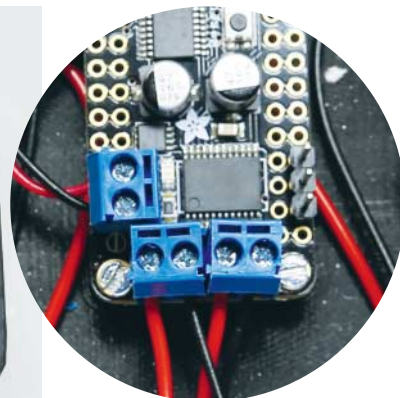
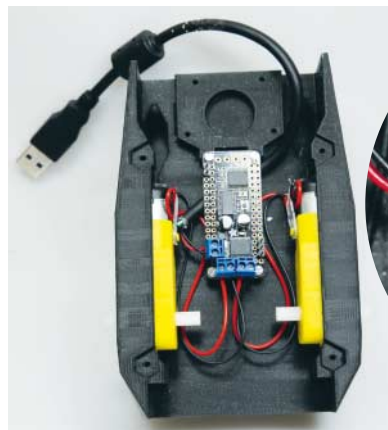
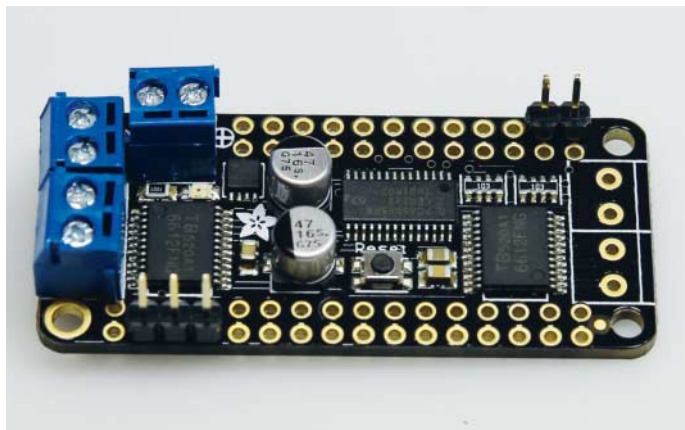
ausführliche Anleitung zum Aufbau der Hardware und zur Software-Installation sowie weitere nützliche Tipps. Das Wiki ist in leicht verständlichem Englisch und bebildert. Wir drucken daher hier keine komplette Bauanleitung, sondern geben nur einige Tipps zum Aufbau.

Falls die beiden Getriebemotoren ohne Anschlussdrähte geliefert werden, löten Sie an jede der beiden Lötflächen der Triebwerke etwa 15cm Litze an, idealerweise in unterschiedlichen Farben, damit Sie beim Anschluss an die Treiberplatine die Polung unterscheiden können.



<sup>1</sup> Das Jetson Nano Developer Kit kann in diesem Roboter ohne Lüfter betrieben werden, da seine Leistungsaufnahme per Software gedrosselt wird.





**2** Nur diese Anschlussklemmen und Pin-Leisten müssen auf der Motortreiber-Platine eingelötet werden. Lassen Sie neben der dreipoligen Leiste einen Anschluss frei.

**3** Die Verkabelung von Motortreiber und Motoren ist einfach. Wichtig ist, dass die Motoren parallel zu den Gehäusekanten sitzen, sonst fährt der JetBot später nur Kurven.

In die Motortreiber-Platine müssen Sie Anschlussklemmen und Pin-Leisten einlöten **2**. Da der JetBot mit nur zwei Motoren arbeitet, reichen die im Bild gezeigten Anschlüsse aus.

Die Anschlussklemmen an der kurzen Seite der Treiberplatine sind für die Motoranschlüsse vorgesehen. Motoren und Platine setzen Sie dann ins Chassis ein **3**. Die Platine wird mit selbstschneidenden 2mm-Schrauben oder normalen 2,5mm-Schrauben befestigt. Die Motorenbefestigung erfolgt mit je zwei M3×30-Schrauben. Wichtig: Die Motoren müssen parallel zur Chassis-Kante sitzen. Falls nicht, säubern Sie das Chassis von eventuellen Resten einer Stützstruktur.

An die seitliche zweipolige Anschlussklemme der Treiberplatine werden die rote und schwarze Ader eines USB-Kabels angeschlossen wie im Bild zu sehen. Darüber erfolgt die Stromzufuhr. Alle Kabel verlegen Sie

so, dass sie später nicht am Boden oder den Rädern schleifen können.

Dann stecken Sie die Teile des Kugellagers zusammen **4** und schrauben es ans Chassis **5**. Achten Sie darauf, dass es nach dem Anschrauben leicht rollen kann. Andernfalls müssen Sie die Anlagefläche des Lagers nacharbeiten.

Das kleine Display, das später die IP-Adressen des JetBot anzeigt, wird über den I<sup>2</sup>C-Bus gesteuert. Der muss aber auch an die Motortreiber-Platine weitergeleitet werden. Daher ist noch eine gewinkelte PIN-Leiste (2 × 3 Pins) an das Display zu löten **6**. Die Kabel für den I<sup>2</sup>C-Bus schließen Sie dann an die Treiberplatine **7** und das Display **8** an.

Zuletzt schrauben Sie das Jetson nano Developer Kit auf das Chassis, stecken das Display auf und schließen die Kamera an. Befestigen Sie die Kamera am Halter **9**. Wichtig: Damit die Stromversorgung des Jetson

über die USB-Buchse erfolgt, muss der Jumper neben dem Kameraanschluss entfernt werden. Jetzt fehlt nur noch die Powerbank. Sie kommt ins Chassis unter den Jetson. Schließen Sie die USB-Kabel des Jetson und der Treiberplatine aber noch nicht an, denn dem Jetson fehlt ja noch die Software.

## Software-Installation

Auch die Software-Installation ist im Wiki gut beschrieben. Das Betriebssystem des Jetson (basierend auf Ubuntu) müssen Sie auf eine Mikro-SD-Karte speichern. Lediglich bei der Software-Aktualisierung im Punkt 5 sind Abweichungen möglich. Hier hilft vor dem `rsync`-Befehl ein

```
sudo apt-get update
```

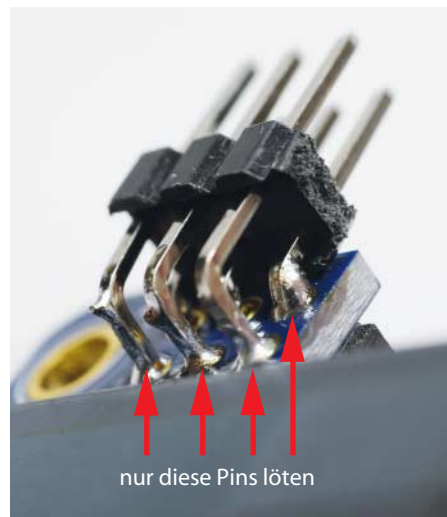
Vergessen Sie auch nicht, die WLAN-Verbindung und den Stromsparmmodus einzustellen.



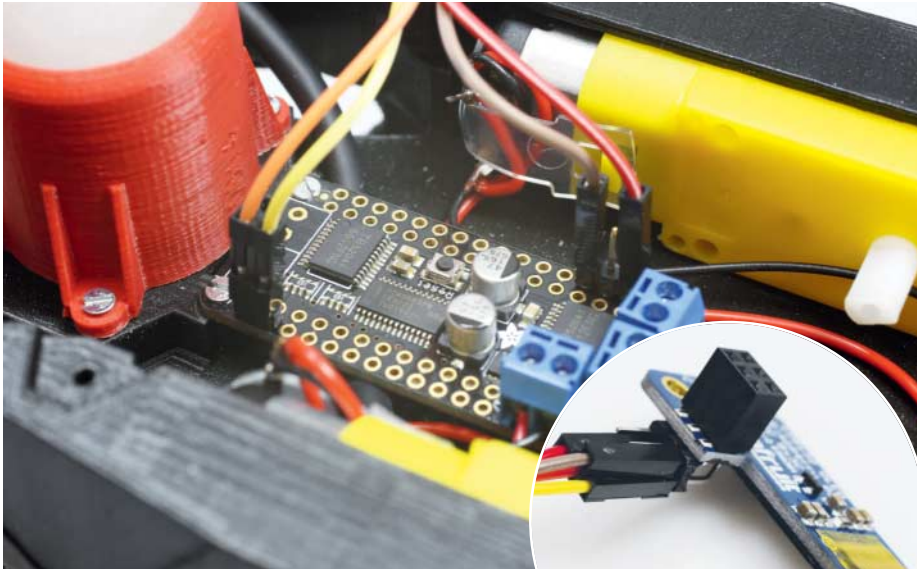
**4** In dieser Reihenfolge müssen Sie die Teile des vorderen Kugellagers zusammenstecken ...

**5** ... und am Chassis anschrauben.

**6** An den Display-Anschluss muss eine gewinkelte Pin-Leiste gelötet werden. Nur die markierten vier Pins müssen gelötet werden, sie sind die I<sup>2</sup>C-Schnittstelle für den Motortreiber.



nur diese Pins löten



7 Die vier Drähte für den I<sup>2</sup>C-Bus schließen Sie am Motortreiber so an ...

8 ... und am Display so.

len, sonst wird die Powerbank schnell leer sein. Die WLAN-Einrichtung nehmen Sie am JetBot vor, daher müssen Sie Monitor, Tastatur und Maus anschließen.

### Web-Oberfläche

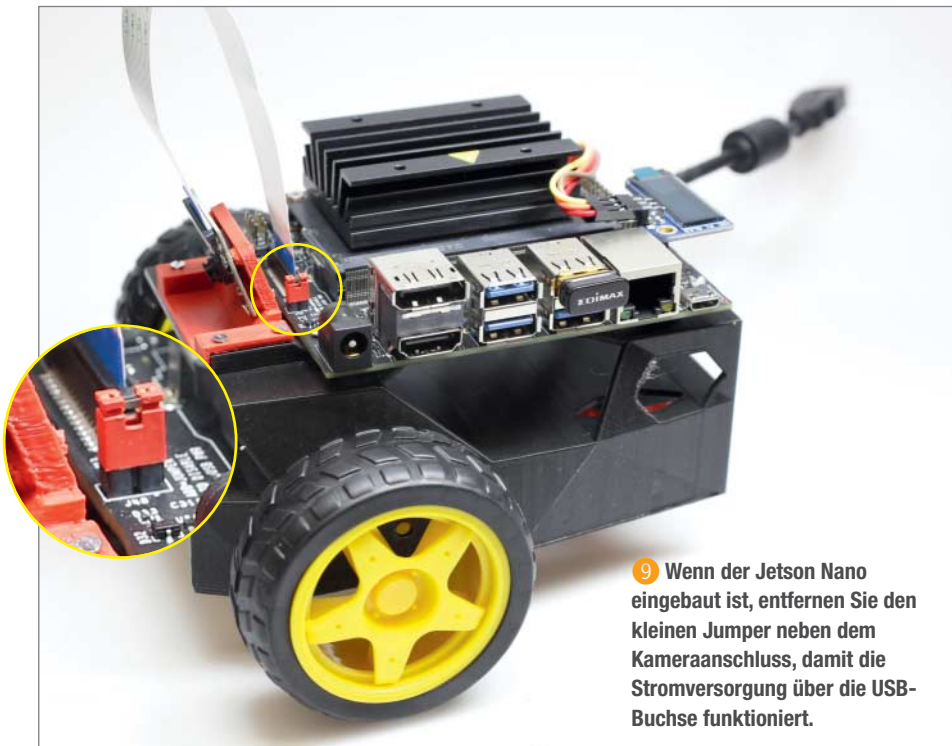
Damit ist der JetBot auch schon einsatzbereit für Ihre und seine ersten Schritte. Verbinden Sie sich von einem Web-fähigen Gerät im

gleichen WLAN wie der JetBot. Geben Sie dazu im Browser diese Adresse ein:

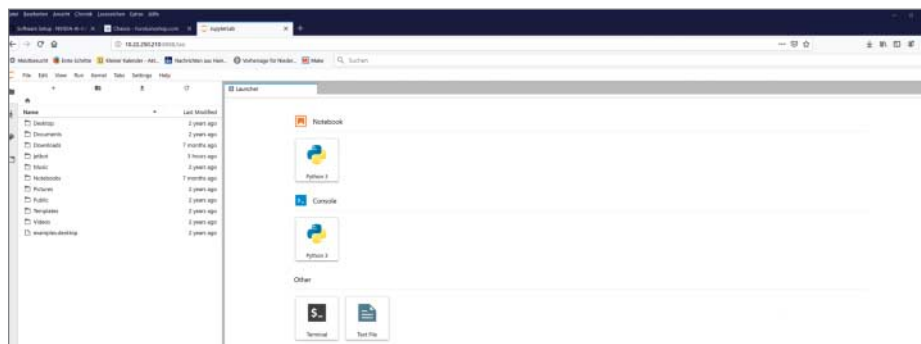
`http://xxx.xxx.xxx.xxx:8888`

für xxx.xxx.xxx.xxx setzen Sie die IP-Adresse des JetBot ein, die er etwa ein bis zwei Minuten nach dem Start auf seinem Display anzeigt. Nach kurzer Zeit werden Sie nach dem Passwort gefragt. Es lautet `jetbot`. Anschließend erscheint die *JupyterLab* genannte

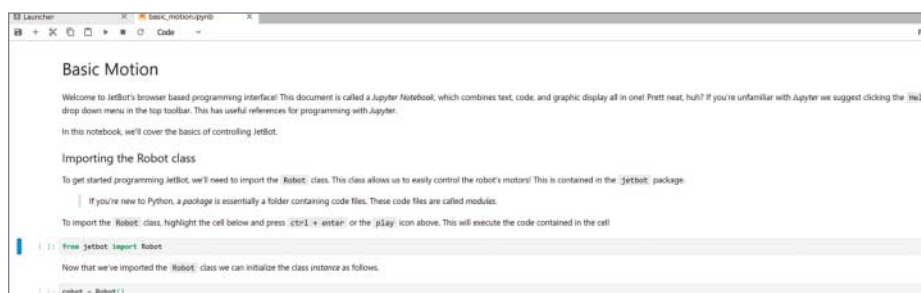
Anzeige



9 Wenn der Jetson Nano eingebaut ist, entfernen Sie den kleinen Jumper neben dem Kameraanschluss, damit die Stromversorgung über die USB-Buchse funktioniert.



## 10 Die sehr aufgeräumte Web-Oberfläche des JetBot



## 11 Nur wenn der blaue Markierungsbalken links neben einem Code-Block steht, kann der ausgeführt werden.

Oberfläche 10. Der linke Teil ist der File-Browser. Hier erscheinen Verzeichnisse und Dateien, die auf dem Jetson gespeichert sind. Im Verzeichnis *Notebooks* liegen die Beispielprojekte. Wählen Sie dort *basic\_motion.ipynb*.

Im rechten Teil des JupyterLab wird nun das Dokument angezeigt. Es handelt sich nicht um ein einfaches Textdokument, sondern es enthält außerdem ausführbaren Programm-Code (grau hinterlegt). Wenn Sie mit der Maus oder den Cursortasten den blauen

Markierungsbalken am linken Rand auf solch einen Code-Block setzen, können Sie die Befehle bei gleichzeitigem Drücken von STRG und Enter ausführen lassen 11.

Beginnen Sie einfach: Lesen Sie die Anleitung und führen Sie die Code-Blocks aus. Achtung: Der dritte Block setzt den JetBot in Bewegung! Beim neunten Block wird Ihnen etwas auffallen: Auch sämtliche Anzeigen, die durch den Programmcode ausgelöst werden, erscheinen im Notebook, zum Beispiel die Schieberegler 12.



## 12 Alle Programmausgaben wie etwa die Schieberegler, aber auch Fehlermeldungen erscheinen direkt unter dem jeweiligen Programmblock.

Die Befehle der Code-Blöcke lassen sich auch ändern: Einfach hineinklicken und die Änderungen eingeben. So können Sie schnell kontrollieren, was das jeweils bewirkt. Spielen Sie einfach etwas herum, so lernen Sie schnell die Bedeutung der einzelnen Anweisungen.

Über das File-Menü können Sie auch selbst eigene Notebooks anlegen und darin Text und Programmcode einsetzen. Sie haben also Raum für viele Experimente.

## Trainieren

Kommen wir nun aber einen Schritt weiter zum autonomen Roboter und bringen ihm das Ausweichen vor Hindernissen bei. Computer sind zwar dumm, aber heutzutage zumindest lernfähig. Deshalb kann und muss der JetBot zunächst trainiert werden, um Objekte zu erkennen, beispielsweise Hindernisse oder freien Bewegungsraum. Dieses Training erfolgt in zwei Teilen: der erste besteht im Anfertigen etlicher Fotos von Hindernissen und unbedenklichen Freiräumen. Es kommt hier vor allem auf eine möglichst hohe Anzahl von Fotos aus unterschiedlichen Perspektiven an. Diese Fleißarbeit müssen Sie mit Hilfe der JetBot-Software erledigen, und zwar mit dem Notebook namens *data\_collection.ipynb*.

Bevor Sie den ersten Programm-Code darin aufrufen, starten Sie zunächst den dazugehörigen Kernel des JetBot erneut 13. Andernfalls kann es zu Fehlermeldungen kommen.

Lassen Sie nun die ersten fünf Code-Blöcke des Notebooks ausführen, also den Markierungsbalken daneben setzen, dann STRG und Enter drücken. Falls der jeweilige Code-Block eine längere Zeit zur Ausführung braucht, erscheint in der eckigen Klammer davor zunächst ein Sternchen. Ist der Code abgearbeitet, wechselt das zu einer Nummer, 1 für den ersten, 2 für den zweiten ausgeführten Code-Block und so weiter.

Unterhalb des fünften Code-Blocks sehen Sie dann das Kamerabild und je einen roten und grünen Kateron 14. Setzen Sie den JetBot nun auf dem Gelände aus, in dem er sich künftig zurechtfinden soll, beispielsweise einen Tisch mit verschiedenen Gegenständen. Anschließend nehmen Sie Fotos auf: Gegenstände und die Tischkanten nehmen sie mit einem Klick auf den roten Button, freie Flächen nehmen Sie mit dem grünen auf. Achten Sie darauf, alle Hindernisse aus möglichst vielen verschiedenen Richtungen aufzunehmen. Das erhöht die Erkennungsgenauigkeit. Sie sollten jeweils etwa 100 Bilder aufnehmen. Den JetBot bewegen Sie dabei von Hand, deswegen ist dies eine Fleißarbeit. Wenn Sie eine ausreichende Bilderzahl erreicht haben,



führen Sie den sechsten Code-Block aus. Damit werden alle Bilder in einem ZIP-Archiv gespeichert.

Der zweite Teil des Trainings beschäftigt vor allem den Jetson nano, denn nun muss das neuronale Netz dort die Bilder auswerten. Da der Leistungsbedarf des Jetson hierbei recht hoch ist, sollten sie ihn nicht über den Akku, sondern per Netzteil mit Energie versorgen (Jumper nicht vergessen!).

Das Training geschieht mit Hilfe des Notebooks *train\_model.ipynb*. Falls das nicht angezeigt wird, finden Sie es im Verzeichnis *jetbot/notebooks/collision\_avoidance*.

Lassen Sie hier alle neun Code-Blöcke ausführen. Es kann eine ganze Weile dauern, bis der Jetson fertig ist. Danach finden Sie eine Datei namens *best\_model.pth* im Notebooks-Verzeichnis. Sie enthält die zur Objekterkennung notwendigen Daten.

## Autonome Bewegung

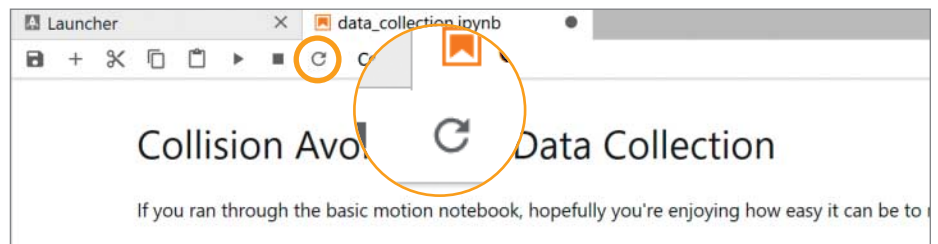
Richtig interessant ist der kleine JetBot aber erst, wenn er sich autonom bewegt. Er hat ja nun auch alles, was er dazu braucht. Entfernen Sie nun die Netzteilverbindung und den Jumper. Nach dem Neustart verbinden Sie sich wieder mit dem JetBot und rufen das Notebook *live\_demo.ipynb* auf. Lassen Sie die ersten acht Code-Blöcke darin ausführen. Einige davon brauchen bis zu einer Minute, bis sie abgearbeitet sind. Der Grund: Jetzt wird auf dem Jetson nano das neuronale Netz für die Objekterkennung entsprechend dem vorherigen Training installiert. Da hierbei eine Menge Daten in den Speicher der GPU zu transportieren sind, dauert das eine Weile.

Schließlich aber wird sich der JetBot in Bewegung setzen. Falls er auf ein Hindernis trifft, weicht er ihm mit einer Links-Kurve aus. Übrigens: Sollte er sich nicht von der Stelle bewegen, weil seine Motoren nicht genug Kraft entwickeln und nur leise summen, können Sie mehr Gas geben. Setzen Sie im siebten Code-Block in den Klammern hinter *roboter.forward(0.x)* und *roboter.left(0.x)* für *x* größere Werte ein (etwa 6 bis 8).

Während der Roboterbewegung wird übrigens das Kamerabild laufend an den Browser übertragen. Das Kamerafenster finden Sie im angezeigten Notebook etwas weiter oben.


Dies ist nur ein kurzer Anriss der Möglichkeiten, die im JetBot stecken. Inzwischen gibt es im Internet schon eine ganze Anzahl an Jet Bot-Programmierern. Einige interessante Adressen dazu finden Sie über den Kurzinfo-Link. Ansonsten haben Sie jetzt das Grundwissen, um selber mit dem Roboter zu experimentieren. Viel Spaß dabei!

—hgb



**13** Bevor Sie den Programm-Codes eines neuen Notebooks starten, setzen Sie den dazugehörigen Kernel mit einem Klick auf dieses Symbol zurück.

```
[5]: display(image)
      display(widgets.HBox([free_count, free_button]))
      display(widgets.HBox([blocked_count, blocked_button]))
```



2

add free

5

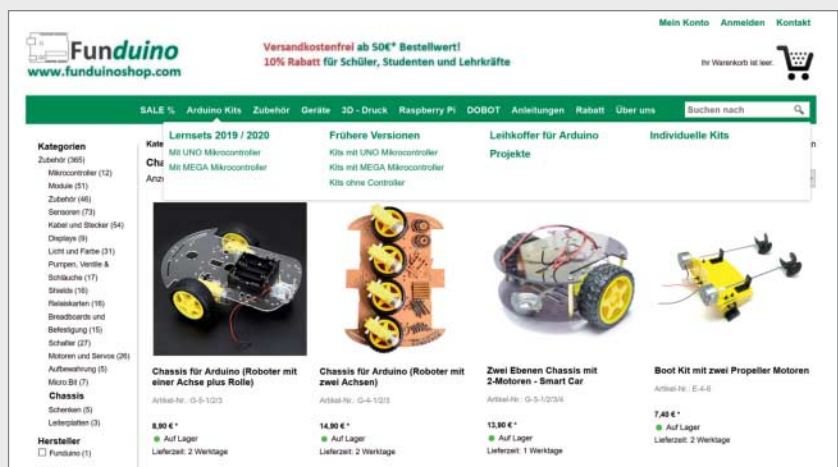
add blocked

**14** Sobald das Kamerabild erscheint, ist der JetBot bereit zur Aufnahme der Trainingsfotos.

## Universal-Chassis als Grundlage

Falls Sie das Chassis des Roboters nicht selber drucken möchten, können Sie auch ein preiswertes Universalchassis verwenden.

Die gibt es im Set für unter zehn Euro bei diversen Internet-Shops. Zur Montage der einzelnen Komponenten des JetBots enthalten sie bereits zahlreiche Montagelöcher. Fehlende lassen sich aber auch schnell nachbohren.



Es muss nicht immer gleich 3D-Druck sein. Auch solche preiswerten Chassis sind brauchbar, erfordern aber eventuell etwas Bohrarbeit.